Task 4 – Blood Donation Forecast (Python Code)

```python
# =========================================
# Task 4: Blood Donation Forecast
# Tech Stack: Python, Pandas, Scikit-learn
# Difficulty: Intermediate
# =========================================

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score

data = {
    "donors_last_month": [120, 150, 90, 200, 170, 130, 160, 110],
    "blood_requests": [140, 160, 100, 220, 180, 150, 170, 120],
    "camps_held": [3, 4, 2, 6, 5, 3, 4, 2],
    "emergencies": [10, 12, 8, 15, 14, 11, 13, 9],
    "donations_next_month": [145, 170, 105, 230, 190, 155, 180, 125]
}

df = pd.DataFrame(data)
print("Dataset Loaded Successfully")
print(df.head())

X = df.drop("donations_next_month", axis=1)
y = df["donations_next_month"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

model = RandomForestRegressor(
    n_estimators=100,
    random_state=42
)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error:", mae)
print("R2 Score:", r2)

future_data = pd.DataFrame({
    "donors_last_month": [150],
    "blood_requests": [165],
    "camps_held": [4],
    "emergencies": [12]
})

future_prediction = model.predict(future_data)
print("Predicted Blood Donations:", int(future_prediction[0]))
```